5.0 DOCUMENTATION ASSESSMENT

The Documentation Assessment portion of ASP-I addresses: the completeness of the available model documentation; the compliance of each component (or volume) of the set to accepted, tailored standards for mature M&S; recommendations as to how the documentation set should be modified to bring it into compliance with those standards; and a listing of implications of the current state of documentation on model use and V&V efforts. This assessment provides the model manager with specific information on how the documents can be improved and also provides the model user with a quick description of the documentation adequacy that could facilitate a decision regarding model usage. The characteristics and adequacy of the model documentation are summarized here using the following terms:

- Completeness — The completeness of the documentation is stated "Complete," "Partially Included," or "Not Included."
- Compliance — The compliance of the documentation with referenced standards is stated as "Complies" or "Does Not Comply."
- Applicability — The version of the model the documentation represents is stated as "Current" (the latest version) or "Version (n.n.n)."

The standards against which the documentation were assessed were derived from a study sponsored by the SMART Project and documented in a report entitled Software Verification Requirements Study (SVRS) for the SMART Project [4]. The SVRS describes the minimum set of documents and content standards required to assist a potential user to evaluate the suitability of an existing model for a specific purpose and ensure that it has been rigorously verified against known standards and procedures. These documents should allow the potential user to: have confidence that the model is accurate; decide if the model simulates the problem(s) of concern; have sufficient information to install and run the program(s); modify the model to work on the target platform (if necessary); understand all inputs and outputs; and fix problems that arise during model use. After an extensive search for and review of government requirements and guidelines, the following documents were identified as the minimum set necessary for mature model verification: Software User's Manual (SUM), Software Programmer's Manual (SPM), Software Analyst's Manual (SAM), Software Design Document (SDD), and Software Verification Report (SVR). Neither a formal SPM nor a SAM exists for TRAP, but the SUM contains much of the information typically contained in these documents. Table 5-1 summarizes the adequacy of the TRAP documentation in the SUM, SPM, and SAM required subject areas.

TABLE 5-1. Summary of TRAP Documentation Assessment.

Characteristic	SUM	SPM*	SAM*
Publication Date	March 31, 1993	March 31, 1993	March 31, 1993
Applicability	TRAP 3.1	TRAP 3.1	TRAP 3.1
Completeness	Does not adequately describe initialization procedures, links to other programs, error messages, or detailed assumptions and limitations.	Does not adequately describe equipment configuration (hardware or software), compiling and linking instructions, or error diagnostics.	Implementation Methodology does not adequately address all simulation modes or provide justification for or impact of algorithms used. High-level and detailed assumptions and limitations not adequately described.
Compliance	Does Not Comply	Does Not Comply	Does Not Comply

* Some SPM and SAM subject areas covered in SUM

5.1 COMPLETENESS

The completeness of a particular subject area is either "Included and Complete," "Partially Included," "Partially Included in Another Manual," or "Not Included." The Software User's Manual for TRAP3.1 [6] contains several volumes and is the only published documentation for TRAP. It is distributed by SURVIAC when requested by the user. The Software User's Manual contains many of the elements included in a standard SUM in addition to many of the elements contained in a standard SPM and SAM. Although the SUM is extensive, it is not adequate to take the place of the SAM and SPM. Table 5-2 summarizes the completeness of each SUM subject area and gives its location in the current document.

TABLE 5-2. Summary of TRAP SUM Content Assessment.

Subject Area	**Status**	**Current Section**
Title Page and Preliminary Information	P	Title Page
1.1 Identification	Y	1.1
1.2 System Overview	P	1.1, 1.2, 2.1, 2.2
1.3 Document Overview	Y	1.3
2.0 Referenced Documents	N	N
3.1 Initialization	P	3.0
3.2 User Inputs	Y	Section 4
3.3 Links to Other Programs	N	N
3.4 Outputs	Y	Section 5
4.0 Error Messages	N	N
5.1 Glossary of Terms	N	N
5.2 Abbreviations	N	N
Appendix A: Detailed Assumptions and Limitations	P	Appendix E, distributed throughout User Manual
Legend Y Included and Complete P Partially Included N Not Included		

5.1.1 Software Programmer's Manual

The SPM does not exist as a separately titled volume; instead some of the required information is included in the SUM. Several required areas, however, are only partially included or not included at all. These discrepancies are detailed in the compliance section below, and are addressed as if the relevant portions of the SUM were part of a SPM. Table 5-3 summarizes the completeness of the SPM information and gives its current location in the SUM.

TABLE 5-3. TRAP SPM Content Assessment Summary.

Subject Area	Status	Current Section
Title Page and Preliminary Information	N	N
1.1 Identification	N	N
1.2 System Overview	N	N
1.3 Document Overview	P:SUM	1.3
2.0 Referenced Documents	N	N
3.1 Equipment Configuration	P:SUM	1.2, 3.0
3.2 Operational Information	P:SUM	1.2
3.3 Compiling and Linking Instructions	P:SUM	3.0
4.1 Programming Information Introduction	P:SUM	1.2
4.2 Call Hierarchy	Y	6.1
4.3 Dictionary of Variables	Y:SUM	Appendices B
4.4 Global Variables	Y:SUM	Appendices B, E
4.5 Program, Subroutine, and Function Descriptions	Y:SUM	Appendix E
4.6 Error Detection and Diagnostic Features	N	N
5.1 Glossary of Terms	N	N
5.2 Abbreviations	N	N
Appendix A: Detailed Call Hierarchy	Y:SUM	2.3.4
Legend N Not Included P:SUM Partially Included in Software User's Manual Y:SUM Included and Complete in Software User's Manual		

5.1.2 Software Analyst's Manual

The SAM does not exists as a separately titled volume; instead, much of the information typically contained in the SAM is included in the SUM. There are, however, several required subject areas that are either only partially included or not included at all. These discrepancies are discussed in the compliance section below, and are addressed as if the relevant portions of the SUM were part of a SAM. Table 5-4 summarizes the completeness of the SAM information and lists its current location in the SUM.

TABLE 5-4. TRAP SAM Content Assessment Summary.

Subject Area	Status	Current Section
Title Page and Preliminary Information	N	N
1.1 Identification	N	N
1.2 System Overview	N	N
1.3 Document Overview	P:SUM	1.3
2.0 Referenced Documents	N	N
3.1 Functional Description Overview	Y:SUM	2.2, 2.3
3.2.1 Assumptions and limitations	P:SUM	Section 7
3.2.2 Overall Modeling Methodology	P:SUM	Section 7
3.3 Detailed description of algorithms	P:SUM	Section 6, 9, Appendix E
4.1 Glossary of Terms	N	N
4.2 Abbreviations	N	N
Appendix A: Detailed Assumptions and Limitations	P:SUM	Appendix E, distributed throughout User Manual

Legend

Y:SUM Included and Complete in Software User's Manual
P:SUM Partially Included in Software User's Manual
N Not Included

5.2 COMPLIANCE

This section presents the compliance of the TRAP documentation set to the set of standards recommended in [4] and [5]. These standards are summarized in the following sections.

5.2.1 Software User's Manual

The purpose of the SUM is to provide the user with enough information to execute the model correctly. The execution steps, expected output, and actions to follow upon encountering an error should be discussed. In addition to providing an introduction to the model, the SUM should provide enough theoretical background information so that the user can gain a deeper understanding of the model and its output.

Standards

The recommended format and contents for a SUM are described in [4] and repeated below:

> **Title Page and Preliminary Information.** A SUM Title Page should include the following information: Model Name, Version Number, Volume Number (if applicable), Development Agency, Contractor Name and Address, Contract and CDRL Numbers (if applicable), Date Published, Distribution and Destruction Notices (if applicable), and Document Control Number (DCN). The term "Prepared by" should preface the listing of the Contractor Name and Address. In addition to the Title Page, a Foreword (Abstract), Table of Contents, List of Tables, and List of Figures should also be provided.

SECTION 1: INTRODUCTION

1.1 Identification. Identify the exact model title, its acronym or abbreviation, the version number, and any other official model identification information.

1.2 System Overview. State the purpose of the model. Include its mission, a general description of the physical systems simulated, and a general description of the intended scenarios. Provide overviews of all major modes of operation and scenarios corresponding to each mode. Auxiliary programs used to generate input data or process output data should be acknowledged; such auxiliary software should be detailed in Section 3.3 (entitled "Links to Other Programs").

1.3 Document Overview. List and describe the purpose of each section of the SUM. Also identify any other documents in the document set containing the SUM.

SECTION 2: REFERENCED DOCUMENTS

List the title, number, author, publisher, date and classification level (unless all are unclassified) for each document used in generating the SUM, and for all known documentation for this model. Include sources for all documents not available through normal government stocking activities.

SECTION 3: EXECUTION PROCEDURES

Present detailed procedures necessary to run the model. The instruction set should be comprehensible by a user unfamiliar with the software design. Each subsection in this section should describe step-by-step instructions for executing the model, including details of the options available to the user at each step.

3.1 Initialization. Describe the initialization procedures necessary to execute the model. Detail all initialization options.

3.2 User Inputs. Describe user inputs at the file or data set level. Include variable name, format, allowable ranges, units of measure, and definition of each input item.

3.3 Links to Other Programs. Detail model relationships with pre- and post-processors. Describe drivers not considered part of the model, but part of the delivered model package. Discuss any other program with a link to a model.

3.4 Outputs. Detail the expected outputs from the model. This includes narrative reports as well as files. When applicable, give filenames with paths, data format and units of measure.

SECTION 4: ERROR MESSAGES

List each possible error message with a detailed explanation of each message. Provide a definitive course of action for each error message, including instructions for restarting the model.

SECTION 5: NOTES

5.1 Glossary of Terms.

5.2 Abbreviations.

APPENDICES. Appendices may be used for ease in document maintenance or for readability of the core text material. Examples of appendix contents are graphs, sample user interface printouts, and any classified information.

APPENDIX A: DETAILED ASSUMPTIONS AND LIMITATIONS

Appendix A is reserved for describing all model assumptions and limitations. These should be organized by major areas of functionality.

SUM Compliance to Standards

This section compares the TRAP SUM to the set of standards listed above. Incomplete sections are described as either "Adequate" for all practical purposes but incomplete in a formal sense or "Inadequate" in both a formal and practical sense.

Title Page and Preliminary Information. Adequate. The title page and Table of Contents comply with the standard. A Foreword, List of Tables, and List of Figures should be generated.

1.1 Identification. Complies. The model title, version number, and acronym are given in Section 1.1.

1.2 System Overview. Adequate. Section 1.1 - Purpose and Section 1.2 - Background state the purpose of the model with a general description of the types of systems simulated. A list of physical systems simulated should be included. Section 2.1 - General Summary and Section 2.2 - Model Applications provide an overview of the major modes of operation with scenarios corresponding to each mode. A list of auxiliary programs used to generate input data or to process output data should be added.

1.3 Document Overview. Complies. Section 1.3 - How To Use This User's Manual describes the purpose of each section of the SUM.

2.0 Referenced Documents. Inadequate. The SUM does not contain a list of references used in creating it, or a list of all known documentation for this model.

3.1 Initialization. Inadequate. The SUM does not contain a detailed description of the initialization procedures required to execute the model. The brief instructions included for IBM Users in Section 3.0 make use of CMS EXECS and are adequate for a user who wishes only to execute the model rather than understand the execution process. The instructions included for Non-IBM Users describe compiling, linking, file naming conventions, and library calls in a general sense. No scripts used to handle input/output files and run the program are described. A step-by-step list of instructions for all of the computer platforms that TRAP can be hosted on should be added to the SUM.

3.2 User Inputs. Complies. Section 4 - TRAP 3.1 Input Files describes user inputs at the file level. Samples with variable names, default values, units, and variable definitions are listed for each data file. Allowable ranges on inputs are also discussed.

3.3 Links to Other Programs. Inadequate. NAIC has developed a number of input data sets for various systems. These data sets are available to users with the proper clearances and should be described in the SUM. Although TRAP does not rely on external post-processing programs, they are often used to interpret model results. A discussion on the interaction of TRAP with other programs and applications should be added.

3.4 Outputs. Complies. Section 5 - TRAP 3.1 Output Files describes each of the files output by the model. Logical units, file contents, and special notes on how to use a particular output or what outputs to expect are provided with example listings.

4.0 Error messages. Inadequate. A comprehensive list of errors and actions to take to correct and avoid them should be added to the SUM.

5.1 Glossary of Terms. Inadequate. A glossary of terms should be generated.

5.2 Abbreviations. Inadequate. Acronyms are used throughout the text and are typically defined where they are first used. Many commonly used abbreviations are not defined (e.g., units of measure). A complete list of acronyms and abbreviations should be included in the SUM.

Appendix A: Detailed Assumptions and Limitations. Inadequate. Appendix E - Detailed Descriptions of Routines cites some assumptions and limitations used in specific routines. Other assumptions and limitations are distributed throughout the text. A single list of all model assumptions and limitations organized by major areas of functionality should be generated and included in the SUM.

5.2.2 Software Programmer's Manual

The purpose of the SPM is to enable a user or programmer to understand the operation of a model; install, maintain, and modify it; and convert it for use on other computer systems. It includes information on compiling and linking the code as well as descriptions of hardware and software requirements and peculiarities. The SPM addresses software implementation issues rather than theoretical considerations.

There is no SPM as a separately titled volume. Thus, by definition, it cannot be in compliance with any standard. This assessment addresses the SPM subject areas contained in the SUM as if they were part of a separate document.

Standards

The recommended format for a SPM is described in [4] and repeated below:

> **Title Page and Preliminary Information.** The SPM Title Page should include the following information: Model Name, Version Number, Volume Number (if applicable), Development Agency, Contractor Name and Address, Contract and CDRL Numbers (if applicable), Date Published, Distribution and Destruction Notices (if applicable), and Document Control Number (DCN). The term "Prepared by" should precede the listing of the Contractor Name and Address. In addition to the Title Page, sections covering a Foreword (Abstract), Table of Contents, List of Tables, and List of Figures should also be provided.
>
> **SECTION 1: INTRODUCTION**
>
> **1.1 Identification.** Identify the exact model title, its acronym or abbreviation, the version number, and any other official model identification information.
>
> **1.2 System Overview.** State the purpose of the model. Include its mission, a general description of the physical systems simulated, and a general description of the intended scenarios. Provide overviews of all major modes of operation and scenarios corresponding to each mode. Auxiliary programs used to generate input data or process output data should be acknowledged and described.
>
> **1.3 Document Overview.** List and describe the purpose of each section of the SPM. Also identify any other documents in the document set containing the SPM.
>
> **SECTION 2: REFERENCED DOCUMENTS**
>
> List the title, number, author, publisher, date, and classification level (unless all are unclassified) for each document used in generating the SPM and for all known documentation for this model. Include sources for all documents not available through normal government stocking activities.

SECTION 3: PROGRAMMING ENVIRONMENT

3.1 Equipment Configuration. Describe the computing devices and operating systems that the model operates on and under (developmental and target environment). List other software required for model execution. An example of a software requirement is a graphical user interface (GUI).

3.2 Operational Information. Describe hardware/operating system characteristics and capabilities required for the model. This includes details such as storage space for the source code with a complete input set, memory requirements with utilization examples, memory protection features and input/output (I/O) characteristics.

3.3 Compiling and Linking Instructions. Present instructions on compiling and linking the model software, and describe equipment needed for such procedures. Detail applicable names and version numbers of equipment or software.

SECTION 4: PROGRAMMING INFORMATION

4.1 Introduction. Describe in general the applicable programming conventions and style used to develop the model. A short development history emphasizing programming style and convention evolution could be helpful for mature models with a diverse history.

4.2 Call Hierarchy. Present a top-level subroutine call tree. It should branch down only as far as the main routines for each major area of functionality. A comprehensive call hierarchy (probably generated by an automated software tool) should be included in Appendix A.

4.3 Dictionary of Variables. List all variables alphabetically and provide a definition of each (with units of measure). State whether each variable is global or local. If global, give the name of the common block containing it. If local, list the module(s) containing it.

4.4 Global Variables. Global variables are contained in common blocks for programs written in FORTRAN and are called external variables for programs written in C. Other programming languages will have their own conventions for the handling of global variables. Using the convention appropriate to the programming language, list these variables alphabetically. For example, the common blocks from FORTRAN programs should be listed alphabetically. For each block, list the variables contained in it, give a general description of these variables, and list the modules in which it appears. For programs written in other languages, just list the variables alphabetically, give a general description of these variables, and list the modules in which they appear.

4.5 Program, Subroutine, and Function Descriptions. Provide detailed information about each program, subroutine, or function (hereafter called "module"). List modules alphabetically. Library functions should be listed but only briefly described. All other module descriptions should contain the following information in a clear, concise format useful to a programmer tasked with maintaining the model.

a. Give a brief narrative description of the module. Its objective and method for fulfilling the objective should be stated.

b. Give its location in a specified file, its call sequence, security classification level, and size (number of lines of executable code).

c. Provide a list of calls made by the module and calls to the module.

d. Alphabetically list all variables used by the module. For each variable, list its dimension, type, usage as input and/or output, engineering units, a very brief description, and its usage as an argument, local, or common variable. The user can refer to the Dictionary of Variables (Section 4.3) for a detailed description.

e. Detailed Description. Elaborate on the objectives and methods used to fulfill the objectives stated in the brief description in list item "a" above. Provide a reference in the SAM if a theoretical discussion related to the modeled processes is provided.

4.6 Error Detection and Diagnostic Features. Describe model error diagnostics. Provide a table listing each error condition, the routine(s) in which it is utilized, the model variable(s)

involved, and the conditions (logic) causing the error. These diagnostics also are summarized in the SUM, Section 4.

SECTION 5: NOTES

5.1 Glossary of Terms.

5.2 Abbreviations.

APPENDICES. Appendices may be used for ease in document maintenance or for readability of the core text material. Examples of appendix contents are subroutine call tree, flow diagrams, sample user interface printouts and any classified information.

APPENDIX A: DETAILED CALL HIERARCHY

Present the complete calling hierarchy in this appendix.

SPM Compliance to Standards

This section compares the TRAP SPM to the set of standards listed above. Again, incomplete sections are assessed as either "Adequate" or "Inadequate" for practical purposes.

Introductory Information. Inadequate. Because the SPM does not exist as a separately titled document, the Title Page and Preliminary Information, Section 1: Introduction, and Section 2: Referenced Documents do not exist and are therefore not in compliance with the standard. Sections 1.1 and 1.2 should be generated using the corresponding sections in the SUM.

3.1 Equipment Configuration. Inadequate. Hardware is mentioned briefly in several locations in the SUM. These descriptions generally reflect the IBM 3081 hardware configuration at NAIC. A brief note to "Non-IBM Users" is included in Section 3. A comprehensive list of hardware configurations, operating systems, and software required for execution should be added. The following is an example of the information that should be included for each system supported:

Hardware (Intel-based Personal Computer)

- Minimum

CPU:	Intel 486DX-33 (1 each)
Memory:	8 Mb DRAM (70ns)
Hard Disk Drive:	200 Mb (1 each)
Floppy Disk Drive:	3.5" 1.44 Mb (1 each)
Disk Controller:	Appropriate for Hard Drive type (IDE, SCSI, etc.)
Graphics Adapter:	VGA or better
etc....	

- Recommended

  As above, but this list would reflect a hardware suite which would "comfortably" handle the processing task, as opposed to "minimally" handling the task.

Software

- Minimum requirements for compilation and execution should be included.
- Additional packages, such as debugging tools, should also be recommended.

FIGURE 5-1. Sample Configuration.

A detailed list of all software required to accomplish a build on each system should be included. This list should include:

- Compiler (with version number)
- Any external library routines required (with version number, if applicable)
- Other utilities (e.g., the DOS memory extender PHAR LAP)
- Any other elements needed that are not part of the basic operating system or compiler (including debugging tools, post-processors, etc.)

3.2 Operational Information. Inadequate. Disk storage requirements for the distributed simulation and its input and output data sets are not cited in the current documentation. Core memory requirements for an IBM 3081 are mentioned in Section 1.2 of the SUM. Memory utilization examples should be included to illustrate requirements for both simple and complex studies.

3.3 Compiling and Linking Instructions. Inadequate. Section 3.0 contains brief installation instructions for IBM and Non-IBM users. The IBM instructions make use of CMS EXECS and are inadequate for a user to understand the process used for a software build. The instructions included for Non-IBM users describe compiling and linking in a general sense. Step-by-step instructions that address compiling and linking on any platform on which TRAP can be hosted should be added to the SUM. Existing TRAP users running the program under different operating systems or platforms than those used at NAIC should be asked to provide their expertise in developing build instructions for their system types.

4.1 Introduction to Programming Information. Inadequate. The SUM does not contain one section that succinctly describes the programming style used to implement TRAP. TRAP is described in Section 1.2 of the SUM as “highly modular” and “flexible”. This flexibility is “accomplished through the use of a user-written top level FORTRAN interface to the TRAP main body of subroutines”. Section 1.2 - Background provides a brief development history that could be used as a basis to describe any programming style and conventions that can be correlated with various stages in the model’s evolution.

4.2 Call Hierarchy. Complies. Section 6.1 contains a top-level subroutine call tree that shows major areas of functionality. A comprehensive call hierarchy is included in Section 2.3.4.

4.3 Dictionary of Variables. Complies. Appendix B - Data Dictionary contains global and local data dictionaries. Global variable entries contain the variable name, units, description, and the common block in which the variable is contained. Local variable entries include variable name, units, description, and the routine where the variable is declared.

4.4 Global Variables. Adequate. Appendix E contains a listing of all common blocks. It does not contain a description of each of the variables in the block, or a list of the modules in which each appears. Much of this information, however, can be obtained in the Global Data Dictionary in Appendix B. Each entry contains the variable name, units of measure,

a one line description of the variable, and the common block in which the variable is contained.

4.5 Program, subroutine, and function descriptions. Adequate. Detailed descriptions of most modules are listed in Appendix E (minor modules whose logic flow and functionality are apparent are not described in detail). Source code listings are included for each module. Each listing contains a header with a brief description of the routine, an outline of the calculations performed, its calling routines, the routines that it calls, its inputs and outputs, a brief description of internally used variables with units, and source code that is broken into numbered blocks. The text following the listing gives a general description of the module followed by a more detailed description of the methodology used in the implementation. The processing flow is described using the numbered blocks from the code listing. The size (lines of executable code) of each module is not explicitly stated.

4.6 Error diagnostics. Inadequate. The SUM does not contain program error handling or diagnostic procedures. After successful installation and compilation, it is recommended that users run the sample case provided and compare their output to the output distributed by SURVIAC with the source code. No other error diagnostic procedures exist. A discussion of how to generate intermediate outputs for debugging purposes should be included.

5.1 Glossary of Terms. Inadequate. A glossary of terms used in the SPM should be added.

5.2 Abbreviations. Inadequate. Acronyms are used throughout the text and are typically defined where they are first used. Many commonly used abbreviations are not defined (e.g., units of measure). A complete list of acronyms and abbreviations should be generated.

Appendix A: Detailed Call Hierarchy. Complies. A comprehensive call tree is included in Section 2.3.4.

5.2.3 Software Analyst's Manual

The purpose of the SAM is to describe the functional structure and algorithms of a model. It should describe the purpose and background of the model in general terms and give detailed technical descriptions of its complete capabilities, structure, and functions. These detailed descriptions should divide the capabilities of the model into the major functions it performs. All equations, algorithms, and decision processes used by each major function should be described in detail. Details should also be given about model assumptions, limitations, and flexibility (e.g., ability to address different types of problems). Inputs and outputs should be described in words rather than file formats. Each module should be described in detail to explain the correlation between the modules and model functional descriptions. The SAM enables the user to understand the theoretical basis of the model and to determine if the model is appropriate for a particular analysis requirement.

The TRAP documentation does not contain a SAM as a separately titled volume. Thus, by definition, it cannot be in compliance with any standard. This assessment addresses the SAM subject areas contained in the SUM as if they were part of a separate document.

Standards

The recommended format for a SAM is described in [4] and repeated below:

Title Page and Preliminary Information. The SAM Title Page should include the following information: Model Name, Version Number, Volume Number (if applicable), Development Agency, Contractor Name and Address, Contract and CDRL Numbers (if applicable), Date Published, Distribution and Destruction Notices (if applicable), and Document Control Number (DCN). The term "Prepared by" should precede the listing of the Contractor Name and Address. In addition to the Title Page, sections covering a Foreword (Abstract), Table of Contents, List of Tables, and List of Figures should also be provided.

SECTION 1: SCOPE

1.1 Identification. Identify the exact model title, its acronym or abbreviation, the version number, and any other official model identification information.

1.2 System Overview. State the purpose of the model. Include its mission, a general description of the physical systems simulated, and a general description of the intended scenarios. Discuss the types of problems addressed and types of answers provided by the model. Provide overviews of all major modes of operation and scenarios corresponding to each mode. Auxiliary programs used to generate input data or process output data should be acknowledged and described.

1.3 Document Overview. List and describe the purpose of each section of the SAM. Also identify any other documents in the document set containing the SAM.

SECTION 2: REFERENCED DOCUMENTS

List the title, number, author, publisher, date, and classification level (unless all are unclassified) for each document used in generating the SAM and for all known documentation for this model. Include sources for all documents not available through normal government stocking activities.

SECTION 3: FUNCTIONAL DESCRIPTION

3.1 Overview. Describe the model's complete functionality without reference to implementation methodology. These descriptions should elaborate on the overall mission and major modes described above in Section 1.2 "System Overview." Descriptions should be presented in the order functional methodologies are described in the sections that follow.

3.2 General Modeling Approach.

3.2.1 Assumptions and Limitations. Describe high-level assumptions and limitations of overall model functionality.

3.2.2 Overall Modeling Methodology. Explain how assumptions, limitations, and the processes involved influence the general modeling methodology.

3.3 Detailed Functional Implementation Methodology. Describe how the capabilities of the model are functionally implemented. Divide this section into subsections corresponding to the model's major areas of functionality; provide the following information for each subsection:

a. Equations and Algorithms. Provide detailed technical descriptions and purposes for use of specific empirical and analytic equations, numerical algorithms, and decision processes used by the function. Use flow diagrams to depict the implemented logic and use illustrations to depict geometrical considerations when appropriate. Justify use of specific probability distributions. When trade-off studies for equation usage were performed, justify use of the chosen equation.

b. Equations for Variables. Present and describe all equations (using mathematical notation) used for calculating variables that are significant in the implementation of the functionality. Indicate the code variable names that correspond with the variables described by these equations.

c. Model Inputs and Outputs. Inputs and outputs relevant to a particular area of functionality should be described in words without reference to code implementation details. Identify the relationship of inputs to the equations and algorithms in one of those areas.

d. Code Module Correlation with Functionality. Identify each module used to implement an area of functionality and describe the processes contained in that module. The description of each module should include its purpose and a detailed technical explanation. Correlate these processes with the model functional descriptions. Applicable library functions may simply be listed with a short description.

e. Impact on Model Results. Describe the impact of the functionality on model results.

SECTION 4: NOTES

4.1 Glossary of Terms

4.2 Abbreviations

APPENDICES. Appendices may be used for ease in document maintenance, examples and illustrations to assist in understanding model capabilities, or for readability of the core text material. Examples of appendix contents are logic flow diagrams, sample user interface printouts, examples of post-processor use, former studies published using this model, and any classified appendices.

APPENDIX A: DETAILED ASSUMPTIONS AND LIMITATIONS

Appendix A is reserved for describing all model assumptions and limitations. These should be organized by major areas of functionality. This appendix is the same as Appendix A of the SUM.

SAM Compliance to Standards

This section compares the TRAP SAM to the set of standards listed above. Again, incomplete sections are labeled either "Adequate" or "Inadequate" for practical purposes.

Introductory Information. Inadequate. Because the SAM does not exist as a separately titled document, the Title Page and Preliminary Information, Section 1: Introduction, and Section 2: Referenced Documents do not exist and are therefore not in compliance with the standard. Section 1 should be generated from the corresponding section in the SUM. Additional information should be added that describes how TRAP output results are impacted by the user's input data uncertainties or approximations. For example, many models may be dependent on wind tunnel data which have inherent uncertainties. The SAM should describe how these data uncertainties propagate through the simulation and affect the solution.

3.1 Functional Description Overview. Adequate. Section 2.2 of the SUM, Model Applications, describes the major simulation modes. Section 2.3 describes the role of each of the vehicles in each type of engagement. Brief descriptions of vehicle properties and subcomponents are included.

3.2.1 Assumptions and Limitations. Inadequate. Section 7 describes high-level assumptions and limitations associated with using point-mass versus 3-, 5-, or 6-DOF simulations. The numerous assumptions briefly described throughout the SUM and its appendices should be consolidated in the SAM. Discussions of each assumption should be expanded to quantify the effect of the assumption on the output of TRAP. For example, in an air launch scenario it is assumed that the missile is in undisturbed airflow. The impact of making this assumption should be fully described in terms of the effect on the simulation

and what, if any, limitations are imposed. Likewise, a comprehensive presentation of the overall limitations of TRAP from the simulation as a whole down to the functional level should be given in the SAM.

3.2.2 Overall Modeling Methodology. Inadequate. Section 7 describes the impact of point-mass versus 3-, 5-, or 6-DOF simulations on the modeling methodology (and fidelity). This should be expanded to include the impact of all major assumptions, limitations, and processes involved on the general modeling methodology.

3.3 Detailed Functional Implementation Methodology. Inadequate. Section 6 - Engagement Modeling describes how a single engagement is functionally implemented in TRAP. Section 6 is divided into 10 subsections. Section 6.1 - Single Engagement Modeling describes the top level routine, FLYOUT, that controls modeling of a single engagement. The types of routines called by FLYOUT can be grouped as either initialization routines, functional models, state update routines, or utility routines. Each of the major routines called by FLYOUT is briefly described.

The initialization routines are described in Sections 6.2 and 6.6. Section 6.2 details the initialization process for the launch platform, target, air intercept radar, and missile. Missile initialization at launch is covered in Section 6.6.

Sections 6.3, 6.4, 6.5, and 6.7 describe the functional implementation of the target, launch platform, air intercept radar, and missile respectively. Each section begins with a description of the routine called by FLYOUT to control a particular function (FLYTGT, FLYAC, FLYMSL, RADAR, etc.). A call tree is provided for each function. Subsections detail the implementation of each of the routines in the call tree. Where appropriate, additional call trees are provided at the subsection level (e.g., Missile Seeker). Branching logic (with variable names) and variable limits are discussed in the text. Flow diagrams are not provided. Where applicable, illustrations are used to depict geometrical properties. While algorithms are described, the exact equations used to implement a given algorithm or area of functionality are not provided. The routines that implement specific areas of functionality are identified. If additional information on a particular routine is required, Appendix E of the SUM contains detailed discussions of most routines with figures, equations, and annotated code listings.

Section 6 merely describes the implementation; no attempt is made to justify the use of chosen equations/models. For example, empirical data is used to determine ground surface clutter in module PCLUT. The origin of the data and the derivation of the empirical function is not adequately explained. Another case in point, the Swerling Model is used to determine the probability of detection. The current description, which consists only of equations, is not adequate for a thorough understanding of this function. While comments embedded in the source code (programmer's domain) reference a report, no description of the model or justification for using it are included in the text (analyst's domain).

Section 6.8 addresses state updates for the target, launch platform, and missile. Again, the logic used to determine whether a particular integration should be performed is included; the equations used for performing the integration are not. Integration methods used are discussed in Section 6.9. Routines that perform particular functions are identified, enabling the analyst to refer to the detailed routine description in Appendix E if necessary.

Section 6.9 describes the four types of utility routines used by the simulation: (1) atmospheric properties, (2) integration routines, (3) coordinate transforms, and (4) routines to access tabular data. Each section gives a brief discussion of the routines used to implement the function. Detailed discussions of each routine are contained in Appendix E.

Section 6.10 discusses user written additions to TRAP. Users may modify subroutine POLICY to model or affect parameters that are not affected by the standard inputs. Examples of typical modifications and hints on how to implement them are provided.

Section 6 presents the implementation methodology of the major areas of functionality but not at the level of detail called out by the standard. Modules used to implement a particular area of functionality are identified, and the purpose of each module is presented. Much of the detailed technical information (equations, specific algorithms, graphs, etc.) are presented in Appendix E at a level appropriate for a programmer. Some of the information in Appendix E, however, is useful to the analyst and should be moved to this section. Inputs and outputs relevant to a particular area of functionality should be discussed. The relationship between input and output parameters should be discussed. Justification for algorithms and equations used should be presented. The impact of the implemented methodology on model results should be assessed and presented.

The Performance Reconstruction Method (SUM, Section 9) should be entirely rewritten for the SAM. The method description should begin with a general discussion of the reconstruction problem followed by an in-depth discussion of the solution and its algorithmic implementation. Additional detailed information on other simulation modes should be added.

4.1 Glossary of Terms. Inadequate. A glossary of terms used in the SAM should be added.

4.2 Abbreviations. Inadequate. Acronyms are used throughout the text and are typically defined where they are first used. Many commonly used abbreviations are not defined (e.g., units of measure). A complete list of acronyms and abbreviations should be generated.

Appendix A: Detailed Assumptions and Limitations. Inadequate. Appendix E - Detailed Descriptions of Routines cites some assumptions and limitations used in specific routines. Other assumptions and limitations are distributed throughout the text. A single list of all model assumptions and limitations organized by major areas of functionality should be generated and included in the SAM.

5.3 RECOMMENDED MODIFICATIONS

The sections that follow describe the changes needed to bring documentation for TRAP 3.1 into compliance with the standards recommended in [4].

5.3.1 Software User's Manual

This section provides recommendations for modifying the SUM. Table 5-5 presents a summary of the recommendations for compliance from the above discussions.

TABLE 5-5. Recommendations for SUM Compliance.

Section/Topic	Recommendations
Title Page and Preliminary Information	Generate Foreward, List of Tables, and List of Figures.
1.1 Identification	None (complies).
1.2 System Overview	Extract from Sections 1.1, 2.1, and 2.2; add list of auxiliary programs.
1.3 Document Overview	None (complies).
2.0 Referenced Documents	Generate from scratch.
3.1 Initialization	Expand Section 3.0.
3.2 User Inputs	Move from Section 4.
3.3 Links to Other Programs	Generate from scratch.
3.4 Outputs	Move from Section 5.
4.0 Error Messages/Action	Generate from scratch.
5.0 Terms and Abbreviations	Generate from scratch.
Appendix A: Detailed Assumptions and Limitations	Extract from Appendix E and locations throughout User Manual.

5.3.2 Software Programmer's Manual

This section provides recommendations for modifying the SPM. Table 5-6 presents a summary of the recommendations for compliance from the above discussion in Section 5.2.2.

TABLE 5-6. Recommendations for SPM Compliance.

Section/Topic	Recommendations
Title Page and Preliminary Information	Generate from scratch.
1.1 Identification	Generate from SUM Section 1.1.
1.2 System Overview	Generate from SUM Section 1.2 once it is brought into compliance.
1.3 Document Overview	Use portions of SUM Section 1.3 applicable to SPM.
2.0 Referenced Documents	Generate from scratch.
3.1 Equipment Configuration	Expand portions of SUM Sections 1.2 and 3.0; include all systems supported.
3.2 Operational Information	Expand part of SUM Section 1.2.
3.3 Compiling and Linking Instructions	Use parts of SUM Section 3; generate step-by-step instructions for all platforms supported.
4.1 Introduction to Programming Information	Generate from portions of Section 1.2.
4.2 Call Hierarchy	Extract from Section 6.1.
4.3 Dictionary of Variables	Move from Appendix B.
4.4 Global Variables	Extract from Appendices B and E.
4.5 Program, Subroutine and Function Descriptions	Move from Appendix E.
4.6 Error Detection and Diagnostic Features	Generate from scratch.
5.0 Terms and Abbreviations	Generate from scratch.
Appendix A: Detailed Call Hierarchy	Move from Section 2.3.4.

5.3.3 Software Analyst's Manual

This section provides recommendations for modifying the SAM. Table 5-7 presents a summary of the recommendations for compliance from the above discussion in Section 5.2.3.

TABLE 5-7. Recommendations for SAM Compliance.

Section/Topic	Recommendations
Title Page and Preliminary Information	Generate from scratch.
1.1 Identification	Generate from SUM Section 1.1.
1.2 System Overview	Use compliant SUM Section 1.2; expand to include types of problems addressed and types of answers provided.
1.3 Document Overview	Generate with portions of SUM Section 1.3 applicable to SAM.
2.0 Referenced Documents	Generate from scratch.
3.1 Functional Description Overview	Extract from Sections 2.2 and 2.3.
3.2.1 Assumptions and Limitations	Extract from Appendix E, Section 7, and various locations throughout User Manual.
3.2.2 Overall Modeling Methodology	Expand information in Section 7; include impact of all major assumptions and limitations
3.3 Detailed Functional Implementation Methodology 1. Equations and Algorithms 2. Equations for Code Variables 3. Inputs and Outputs 4. Module Correlation with Functionality 5. Impact on Model Results	Expand to cover all simulation modes. Reorganize Section 6, extract information from Appendix E. Add justification for equations/algorithms used. Add flow diagrams where appropriate. Present equations used for calculating significant variables that represent specific functionality. Describe inputs/outputs to particular functions. Describe impact of individual functions on overall model results.
4.0 Terms and Abbreviations	Generate from scratch.
Appendix A: Detailed Assumptions and Limitations	Extract from Appendix E and locations throughout the User Manual.

5.4 IMPLICATIONS FOR V&V

The quality of the current TRAP documentation is assessed to be good for the information included. Extensive reformatting will be needed to achieve standardization; yet this does not significantly impact V&V activities for the information included.

The inclusion of information currently missing will facilitate V&V efforts in the future. In particular, much of the excluded information intended for use by the analyst is instrumental in the verification process. A comprehensive discussion of assumptions and limitations is needed to completely describe theoretical considerations of the modeled processes and to ensure that the model is applied correctly and is appropriate for a specific purpose. In addition, verification efforts will be impeded by a lack of documented sources and justification for equations and algorithms used.

5.5 IMPLICATIONS FOR MODEL USE

The TRAP documentation does not consist of separate User's, Programmer's, and Analyst's Manuals as called out by the standard. In many cases, all three functions are performed by a single person, and a combined manual such as the one provided may be sufficient. Still, a distinction between using a simulation, programming it, and analyzing its output is useful, and a TRAP user may spend hours searching through the extensive

documentation package provided to find an answer in his area of concern. The quality of the information contained in the manual is good, although some required subject areas are missing and there is room for expansion in others. The implication of these deficiencies is addressed in the following paragraphs.

The purpose of the SUM is to provide enough information to run the model correctly. Execution procedures, expected outputs, and actions to perform when errors are encountered should be addressed. The execution procedures provided are sufficient for experienced users who are very familiar with their computer systems and very comfortable with running models and simulations in general. The instructions provided are not sufficient for inexperienced users. A step-by-step list of initialization procedures from setting up input files to executing the program for any platform on which TRAP can be hosted should be included. Information on the input data sets available from NAIC should be included in the manual so the user does not spend a great deal of time generating these inputs with data from scratch. TRAP does not output many error messages, but a list of messages and actions to take to avoid and/or correct them would be helpful to the novice user. A single list of all model assumptions and limitations organized by major areas of functionality should be included so a user can make a quick-look determination of whether the process of concern is modeled at the level of detail required for his/her purpose -- in other words, whether to run the model or not. This information is included in Section 3 of this document, which should be used in conjunction with the model documentation set.

The SPM should enable a programmer to understand the operation of the model, to install and maintain it, to modify it for his/her purposes, and to convert it for use on other platforms if necessary. The information provided in the Program, Subroutine, and Function Descriptions section is adequate for most users to understand the operation of the model. The information provided on installing, maintaining, and modifying the model, however, is sufficient only for an experienced user very familiar with his particular platform and M&S in general. The compiling and linking instructions provided are general. Step-by-step instructions for each platform supported should be provided. In addition, the basic hardware and software requirements for hosting the simulation are not adequately described. Although TRAP does not provide a direct means for diagnosis of errors, a discussion on how to generate intermediate outputs for debugging purposes should be included.

The SAM should provide insight into the functional structure of the model and its algorithms. Concise lists of high-level and detailed assumptions and limitations, and the impact of those on the implementation methodology and model results should be included in the SMA but can be found in Section 3 of this document. In addition, without justifications for algorithms used and a list of sources, the analyst cannot make a determination of whether the implemented approach is the best approach or look elsewhere for an understanding of the implementation selected. The level of detail of the description of single engagement modeling is good and should be applied to the other simulation modes.

While the information contained in the combined manual is good, it is not covered at the level of detail called out in the standard, and the user must study the source code or spend a great deal of time in discussions with the simulation developer to gain a full understanding of some of TRAP's capabilities. This point was mentioned repeatedly in History Research Questionnaires (see Section 4) and in personal discussions with TRAP users. Completing the documentation set would not only get users up and running sooner, it would help them produce more consistent results.